C a l i d a d d e s o f t w a r e

**Factores y características del sistema**

Brenda Juárez Santiago

PROFESORA

Quinto Cuatrimestre

**ÍNDICE**

[IDENTIFICAR LOS FACTORES Y CARACTERÍSTICAS QUE DETERMINAN LA CALIDAD DEL SOFTWARE, COMO: 3](#_1fob9te)

[Funcionalidad 3](#_3znysh7)

[Corrección 3](#_2et92p0)

[Confiabilidad 4](#_tyjcwt)

[Eficiencia 4](#_3dy6vkm)

[Usabilidad 4](#_1t3h5sf)

[Mantenibilidad 5](#_4d34og8)

[Portabilidad 5](#_2s8eyo1)

[Robustez 5](#_17dp8vu)

[Compatibilidad 5](#_3rdcrjn)

[BIBLIOGRAFÍA: 7](#_26in1rg)

**IDENTIFICAR LOS FACTORES Y CARACTERÍSTICAS QUE DETERMINAN LA CALIDAD DEL SOFTWARE, COMO:**

**Funcionalidad**

* Que el usuario pueda utilizar el software

Software de aplicación: en última instancia, todos los elementos se utilizan para que el usuario de cada estación, pueda utilizar sus programas y archivos específicos. este software puede ser tan amplio como se necesite ya que puede incluir procesadores de texto, paquetes integrados, sistemas administrativos de contabilidad y áreas afines, sistemas especializados, correos electrónico, etc. el software adecuado en el sistema operativo de red elegido y con los protocolos necesarios permiten crear servidores para aquellos servicios que se necesiten.

 (Acalle, 2016)

**Corrección**

* Que los datos sean íntegros

Comenzaremos con algunas breves definiciones sobre lo que es corregir:

Es la acción de rectificar, enmendar los errores o defectos de alguien o algo.

Enmendar lo errado. Advertir, reprender.

Un defecto en un programa es una característica con gran potencial de provocar un fallo y este fallo puede llegarse a producir cuando el funcionamiento de un sistema no es el esperado, con respecto al funcionamiento antes definido por las especificaciones del cliente.

Las fallas en un programa suelen ocurrir por diversos motivos pero los más comunes pueden ser:

* En el procesamiento: Es decir se obtienen salidas que no eran las esperadas por el usuario.
* En el rendimiento: Es cuando el tiempo de respuesta del programa no es tan rápido como se esperaba.
* En la programación: Este punto se refiere a cuando existen una serie de inconsistencias en cuanto al diseño.
* En la documentación: Se refiere a las incongruencias que se hayan entre la forma de funcionar del sistema y el manual creado para el usuario.

(Software, 2016)

**Confiabilidad**

* Fácil de usar, fácil de aprender a usar

La confiabilidad es un atributo que mide el grado en que un producto opera sin fallas bajo condiciones establecidas por un periodo de tiempo determinado. La confiabilidad es un atributo cuantitativo que ha sido ampliamente analizado, estudiado y usado en otras industrias para caracterizar la calidad de los productos o servicios. En su concepción más general, la confiabilidad es un atributo que mide el grado en que un producto opera sin fallas bajo condiciones establecidas por un periodo de tiempo determinado. (Buzz, 2016)

**Eficiencia**

* La eficiencia del software está dada por la forma en que hace su trabajo de una manera buena.

¿Cuáles son los recursos que necesita el sistema?

La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

Consiste en hacer el mejor uso posible del espacio de memoria disponible, a la vez que se consiguen las ejecuciones más rápidas. La eficiencia también es un aspecto interno, ya que hay algunos aspectos, (como algunos casos de ocupación temporal de memoria) que no pueden ser detectados por el usuario. (facil), 2016)

**Usabilidad**

Sería el nivel superior. No sólo un software debe hacer lo q e dice q e hace también debe hacer lo que dice que hace; también debe permitimos hacerlo de forma adecuada, natural. (Jalisco, 2016)

**Mantenibilidad**

* Se refiere a la manera fácil de corregir y reparar fallas que pueda tener algún software.

(facil), 2016)

**Portabilidad**

Este factor es de gran importancia porque se refiere a la manera en que los clientes pueden acceder a los productos ya que un software portable es mucho más fácil de obtener por los clientes  dado que pueden acceder a dicho software.

El esfuerzo requerido para transferir un producto de software de una plataforma (entorno de hardware y software) a otra.

Es la posibilidad de pasar de un programa de un ordenador a otro diferente. También se puede entender como la posibilidad de pasar un programa diseñado bajo un S.O., a otro, lo cual es mucho más difícil cuanto más pronunciadas sean las diferencias entre ambos S.O. (piénsese p.ej. en el traspaso de un programa bajo DOS, al entorno gráfico de Windows). Para ello, es conveniente que los programas no hagan uso directo de las características propias de cada máquina o S.O.

(facil), 2016)

**Robustez**

Es la capacidad del producto software para poder funcionar incluso en condiciones fuera de lo normal. Un caso típico son los Sistemas en tiempo real, y los Sistemas tolerantes a fallos. P. ej., cuando en Windows aparece el Error de protección general se trata de una falta de robustez, puesto que, aunque el programa no se rompe, sí impide proseguir la tarea, sin proponer ninguna solución ni intentar auto-solucionar el problema por sí mismo. Esto es inadmisible p.ej. en un sistema de piloto automático en una aeronave.

(algebraicas, 2016)

**Compatibilidad**

Es la facilidad de los programas para combinarse entre sí. También se puede ver como la posibilidad de usar los resultados de un programa como entrada a otro. La clave de la compatibilidad es la homogeneidad en el diseño y un consenso en las convenciones sobre estandarización para las comunicaciones entre programas; en definitiva, la reutilización de los formatos de bloques de información. Esta es una de las principales ventajas del entorno Windows, que inicialmente permitía el transporte de información de una aplicación a otra a través del portapapeles; esto se vio potenciado enormemente con la incorporación de la tecnología OLE. (algebraicas, 2016)

**BIBLIOGRAFÍA:**

Acalle, A. (20 de Enero de 2016). Funcionalidad. Obtenido de Funcionalidad: http://albertolacalle.com/hci/funcionalidad-usabilidad.htm

algebraicas, S. s. (21 de Enero de 2016). Temas 1 y 2 Sistemas software y especificaciones algebraicas. Obtenido de Temas 1 y 2 Sistemas software y especificaciones algebraicas: http://www.lcc.uma.es/~galvez/ftp/tad/Tema01-02.pdf

Buzz, S. (20 de Enero de 2016). Conocimiento para crear software grandioso. Obtenido de Conocimiento para crear software grandioso: http://sg.com.mx/content/view/271

facil), T. z. (21 de Enero de 2016). Factores Que Determinan La Calidad De Un Software? Obtenido de Factores Que Determinan La Calidad De Un Software?: http://tuzonasistema.blogspot.mx/2014/07/factores-que-determinan-la-calidad-de.html

Jalisco, U. T. (20 de Enero de 2016). Calidad en el desarrollo de software. Obtenido de Calidad en el desarrollo de software: http://ozarate.net/material/calidad\_presentacion1.pdf

Software, I. d. (20 de Enero de 2016). Ingenieria de Software. Obtenido de Ingenieria de Software: http://clases3gingsof.wikifoundry.com/page/De+Correcci%C3%B3n